FIT3077 Sprint One
Team: The Honours

Alex Ung, Romal Patel, Hirun Hettigoda

Table of Contents

1) Team Information

Team Name and Team Photo

Team Name: The Honours

Team Membership

Name	Email
Alex ung	aung0008@student.monash.edu
Romal Patel	rpat0035@student.monash.edu
Hirun Hettigoda	hhet0002@student.monash.edu

Name	Technical and or Professional Strengths
Alex ung	- Good communicator - For the most part I have good Organisational skills - I ask a lot of questions - Proficient in coding
Romal Patel	- Fullstack development - Well organised - Skilled in quality testing
Hirun Hettigoda	- Working and communicating well

	within a team - Time management - Quick learner - Frontend and backend knowledge in variety of languages and frameworks

Name	Interesting Fact
Alex ung	I caught a 4kg rainbow trout.
Romal Patel	Champion in Rainbow Six Siege
Hirun Hettigoda	Dropped 2 nukes in mw3

Team Schedule and Workload

The team will utilise "when2meet" to organise each team member's availability. The team will have a weekly in person meeting on Monday (unless otherwise) that will consist of discussing and breaking down deliverables into tasks to be completed. These tasks will be recorded and updated through our "tasks" channel, available in the teams "discord" group, which will also be our main source of communication. If applicable, these tasks will be distributed equally through difficulty and quantity, favouring those with the skill sets for that particular task. If inapplicable, tasks will be completed together as a team through online meetings or in person.

Technology Stack and Justification

The team has decided to utilise Java as the main programming language for this project. The reason for this decision is due to the fact that Java is naturally more strict with object-oriented programming (OOP) which we believe will assist us in keeping in line OOP standards.

Although Python is generally considered more user-friendly for coding applications due to its high-level nature, less stringent syntax, and readability, it offers less control over object-oriented aspects. For instance, errors that would typically be flagged in Java might not be detected in Python.

For the development of the game, we will be utilising JavaFX as it provides all the tools needed to develop a complete application with modern UI elements and features. Furthermore, applications written on JavaFX's library can consistently run on multiple

platforms making the game highly portable. Before finally selecting JavaFX the team was also considering Java Swing as it was also a good library choice with lightweight components. However with the modern building features, 3D graphics and CSS styling that JavaFX provides we believe that it would provide a better result for this project. Overall, the team is approaching JavaFX as a new and exciting challenge as no team members has had previous experience with this. However, we are highly optimistic on the benefits this challenge will reap.

2) User Stories

Submit a list of 20-25 user stories that covers the basic fiery dragons gameplay and
Any extensions that we may want to add. Majority of the user stories should be based on basic requirements

1. As a player I would like to have an indicator of whose turn it is to flip a chit card so that someone doesn't take up my turn.
2. As a player I want to be able to choose how many players there are in the game so that I can play with a single friend or with 3 of my friends.
3. As a player I want to see where my dragon token is so that I can know how far I am from the finish line.
4. As a player I want to see the animals on the volcano card so that I know what animal I need to move forward.
5. As a player I want to see a congratulations screen so that I know who has won the game.
6. As a player I want to see the chit card flip over when I click on it so that I can see what's under the card.
7. As a player I want to see a leaderboard so that I can see who is in the lead during a game.
8. As a player I want there to be a help button so that I can read the rules to understand the game.
9. As a player I would like to select my player icon according to the preferences so that it is unique and avoids confusion when the game has started
10. As a player if I selected correctly on my current turn I would like the chit card to stay turned around so that I can avoid selecting the same chit card on my next turn
11. As a player I would like the chit cards to remain in the same position throughout the game so that the game runs in a fair and smooth manner. (unless extension changes this)
12. as a player I would like to see a "start" menu with a "play" button prompting me to begin so that I as well as the other players are aware when the the game has started
13. as a player I would like a pause menu during the game so that I can quit to the "start" menu or put the game on hold when I desire
14. As a player I would like the game to operate in a clockwise direction so that it is fair and doesn't cause any confusion (unless extension changes this factor)

15. As a player I want a sound to play when the player has flipped a chit card so that the player along with competitors are aware a move has been made
16. As a player I want a sound to play when the player has flipped a successful chit card to move forward so that the player along with the competitors are aware that a correct chit card has been flipped
17. As a player I want the number of caves be equal to the maximum number of players in the game so that each player has a starting and finishing destination (in the basic game this is 4)
18. As a player I want to start the game in a cave which is unique from other players so that there is no confusion in whose turn it is as well as having a unique start and finish destination
19. As a developer I want to make the code extendable so it is easy to add new and different features to the game.
20. As a player I want a mode where the chit cards are randomised after each draw so players cannot memorise where each card is.
21. As a player I want a timer for each player's turn where if they don’t make their turn in time it randomly selects a chit card for them so that the game can continue running at a good pace.
22. As a player I want to be able to select my colour for my animal from a set of contrasting colours so it is easy to see and distinguish between each player.
23. As a player I want a mode where I can extend the map with more volcano cards so that I can have longer games.
24. As a developer I would like to add music to the game to have a nice ambience to the game
25. As a developer I would like the game to run on multiple platforms in order to give users more accessibility.

3) Domain model Diagram

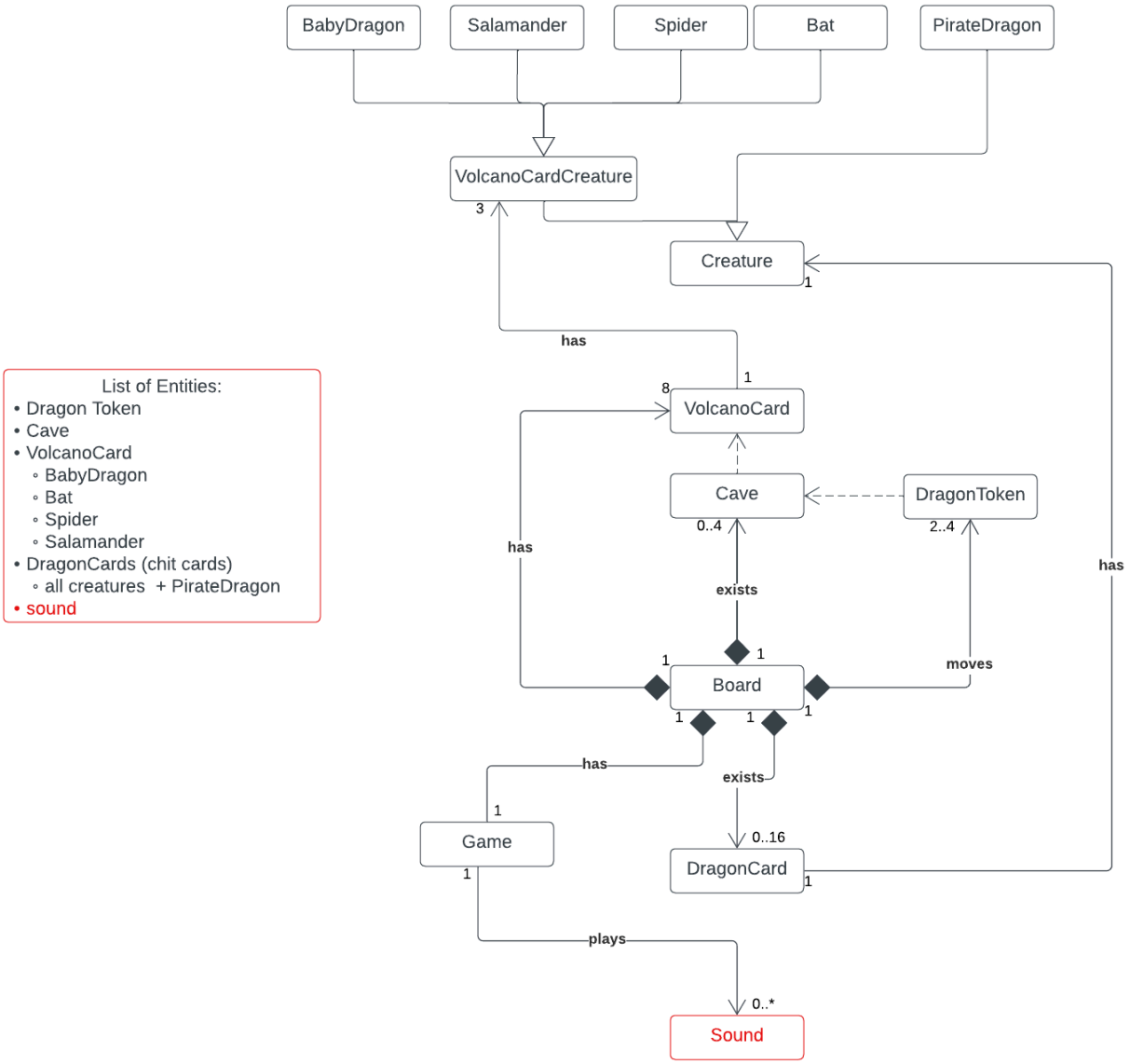


Fig. 1. Final version of Fiery Dragons Domain Model.

Detailed justifications for domain model diagrams

- The domain model consists of a Volcano card creature entity. This entity illustrates the different types of creatures which can be displayed on a volcano card. At this moment in time, the game consists of a Baby dragon, Salamander, Spider and a Bat.

- These creatures are also their own entities which have a generalisation relationship with the VolcanoCardCreature entity since they are a type of creature but not all creatures are VolcanoCardCreatures.

- If more creatures were to be represented as a volcano card creature this can be done so in the future by adding entities to the VolcanoCardCreature.

- The VolcanoCardCreature entity has a generalisation relationship with the creature entity since it is a type of creature.

- There is also an association relationship between the VolcanoCard entity and the VolcanoCardCreature entity to illustrate that 1 volcano card has 3 volcano card creatures displayed on it. The number of volcano card creatures on a volcano card can be changed in the future with extensions if necessary.

- There is a Creature entity. It is a more broad entity to represent all creatures within the game.

- The creature entity was created separately to account for other creatures like the pirate dragon which is a unique entity with no connection to VolcanoCardCreatures.

- The pirate dragon entity has a generalisation relationship with the creature entity since they are a unique type of creature.

- DragonCard has an association relationship with the creature entity in which 1 dragon card represents 1 to 3 creatures as each dragon card can have 1 as a minimum and 3 as a maximum number of creatures on them. We considered having a 1-1 creature relationship because of the perception that a chit card has only one type of creature, however we did not choose to pursue this due to inconsistencies with the logic.

- The DragonCard takes into account all types of creatures, in this case both volcano card creatures and pirate dragons, this is the reason as to why it is connected with the creature entity and not only the volcano card creature entity. This allows for the ability to include more variety of chit card outcomes in the future extensions of the game.

- Furthermore, the Board entity is compromised with multiple composition relationships that highlight that the Board is the overarching entity that will contain

the elements on the board, such as the volcano cards that make up the route, the cave cards which highlight the start and end locations of the players, the DragonToken which the player moves and the DragonCards which the player unfolds. The Board has a composite relationship with these entities as without the Board, these entities do not have a place to exist.

- The DragonToken also has a dependency relationship with Cave as each DragonToken will have a specific cave to start and end at. Hence, the Dragon token will need to fetch the location of the Caves.

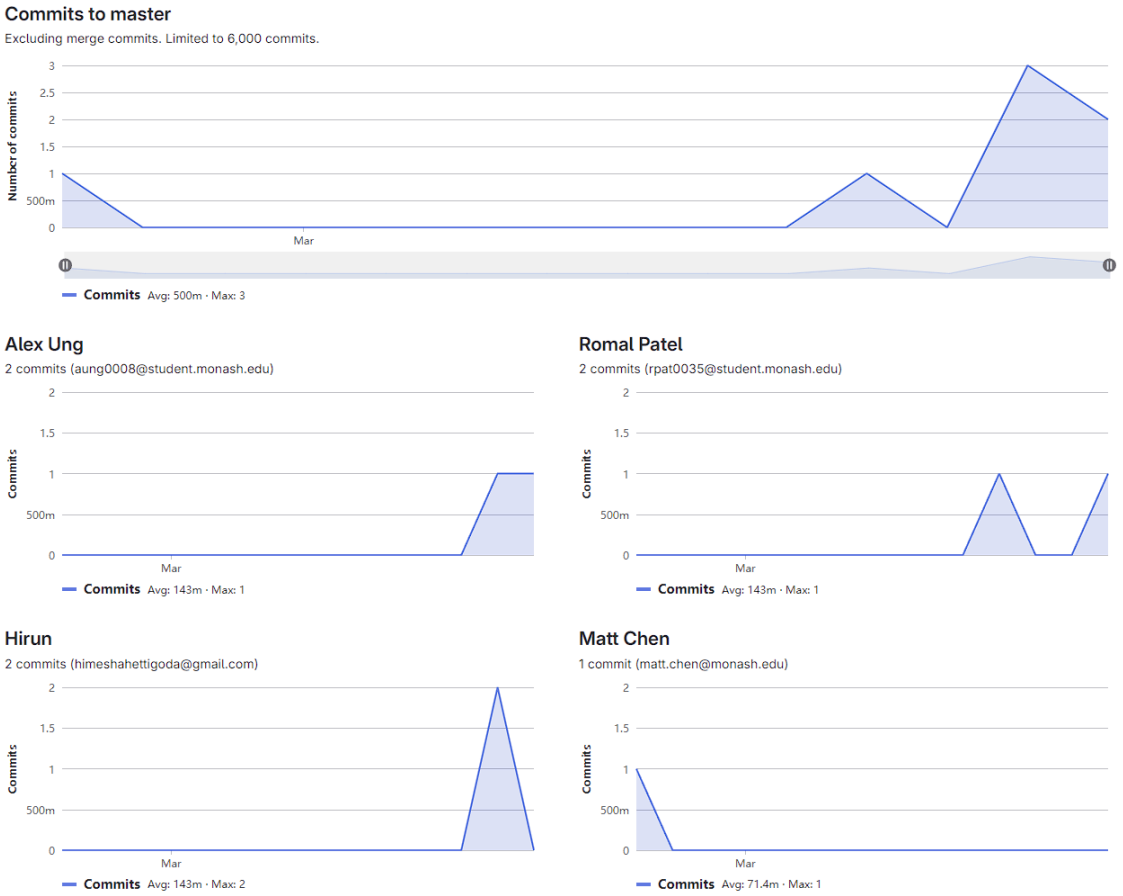
- In addition, the introduction of the sound entity (although not integral to the base functionality to the game) is to account for future extension of the domain model. It is associated with the game because the game is indirectly connected to all entities that require sound. For example, sound is required to flip chit cards, for movement of dragon tokens, shuffling cards, positioning volcano cards and so on. Initially there were unique entities like CorrectMatch or FlippedDragonCard which had a generalisation relationship with the sound entity. However since the game can have 0..* sounds, we believe that unique entities like CorrectMatch should be removed leaving only the sound entity having an association relationship with the game entity as represented in the domain model.

4) Basic UI Design

Low-fidelity prototype with basic dragon token starting positions as well as all chit cards flipped over to show the value of each chit card.

Contributor Analytics

Please note that although we've added a screenshot of the contributor analytics, this is not an accurate representation of the contribution each member for sprint 1 as we did not use commits in the earlier stages of the sprint.



Wiki contribution Log

The wiki contribution log presents a more accurate identifier of contribution. For a more in-depth contribution history, please refer to the version history of the google document linked below:
https://docs.google.com/document/d/1J89LU4yRYZzJJhWHKDx-j1K0cy-MRN_1jv05V273r54/edit?usp=sharing

Sprint 1 Contribution Log

Team Contribution Log

Name	Task
Romal	Created Domain model and implemented main entities and their relationships
Alex	Added user stories 1-8
Alex	Added the creature and related entities and relationships into the model
Romal	Described teams schedule and how the workload will be distributed in the documentation
Hirun	Added the justification of creatures, volcano card creature and chit cards for domain model
Hirun	Added the sound entity along with the relationship between specific sound and game
Hirun	Created some of the volcano cards which are represented in the basic UI design
Hirun	Added user stories 8-17
Romal	Added user stories 17-25
Romal	Added Team Stack and Justification
Hirun	Added info about JavaFX vs Java Swing on tech stack
Alex	Formatted Sprint 1 Documentation
Alex	added low-fidelity prototype that was collectively created by the team during the workshop
Alex	Adjusted some of the domain model justifications